



TITLE:

Rational Runge-Kutta Methods(Numerical Ordinary Differential Equations and Related Topics)

AUTHOR(S):

大野, 博

CITATION:

大野, 博. Rational Runge-Kutta Methods(Numerical Ordinary Differential Equations and Related Topics). 数理解析研究所講究録 1993, 841: 183-201

ISSUE DATE:

1993-05

URL:

<http://hdl.handle.net/2433/83541>

RIGHT:

Rational Runge-Kutta Methods

千葉大・工 大野 博 (Hiroshi Ohno)

1. はじめに

初期値問題

$$y' = f(y), \quad y(x_0) = y_0 \quad (1.1)$$

の数値解法を考える。この中で、数値的に解きづらい硬い微分方程式系と呼ばれるものがある。Wambecq は硬い微分方程式系の数値解法の一つとして、有理型ルンゲ-クッタ法

$$\begin{cases} y_{n+1} = y_n + h \frac{\sum_{i=1}^s \sum_{j=1}^s w_{ij} g_i g_j}{\sum_{k=1}^s c_k g_k} \\ g_i = f(y_n + h \sum_{j=1}^{i-1} b_{ij} g_j), \quad i=1, 2, \dots, s \end{cases} \quad (1.2)$$

と、 g_i がベクトルでも計算できるように、ベクトル演算

$$\frac{ab}{d} = \frac{a \cdot \text{Re}(b d) + b \cdot \text{Re}(d a) - d \cdot \text{Re}(a b)}{(d, d)}, \quad (,) \text{ は内積} \quad (1.3)$$

を提案した[4]。硬い微分方程式系を効率的に解くには、Dahlquist が提案したA-安定性が必要である。線型ルンゲ-クッタ法

$$\begin{cases} y_{n+1} = y_n + h \sum_{i=1}^s w_i g_i \\ g_i = f(y_n + h \sum_{j=1}^{i-1} b_{ij} g_j), \quad i=1, 2, \dots, s \end{cases} \quad (1.4)$$

をA-安定にするためには、慣習として、微分方程式系に対しても使うのに単一微分方程式($y' = \lambda y$, $\text{Re}(\lambda) < 0$) をモデルとして係数 b_{ij} , w_i を決めている。そこで、Wambecq も慣習に従って単一微分方程式をモデルとして係数 b_{ij} , C_k , w_{ij} を決め、A-安定となる公式があることを示した。

しかし、Hairer は Wambecq が提案した公式を使って硬い微分方程式系を解くと、線型ルンゲークッタ法と同様に効率的に解けないことに気づいた。この原因は、慣習として、公式のA-安定性を決めるのに単一微分方程式を使っていたためである。そこで、Hairer は Dahlquist の提案した安定性を厳密化した定義を提案し、A-安定性を弱めた公式が得られることを示した[2]。

公式の安定性は、テスト方程式 $y' = A y$ の数値解の挙動を規定したものである。Hairer の提案した安定性は、解のユークリッド・ノルムに関して規定したものである。線型ルンゲークッタ法は、解の挙動をどんなノルムでみても成分でみても、A-安定性を満たす公式がある。これに対して、Sottas は、有理型ルンゲークッタ法の解の挙動を成分でみるとA-安定性を弱めた公式がないことを示した。このことから、彼は、有理型ルンゲークッタ法が硬い微分方程式系に適さないことを主張している[3]。ただ、解の挙動を成分でみた場合にA-安定性

を満たしていないと、硬い微分方程式系に適していないという根拠がはっきりしない。

以上より、線型ルンゲークッタ法の場合では、(i)微分方程式系に使うのに、単一微分方程式で安定性を決めたり、(ii)数値解のノルムの挙動なのか成分の挙動なのかをはっきりさせなくても、矛盾はでなかった。しかし、有理型ルンゲークッタ法のように非線形なものが入りこんでくると、上記(i)、(ii)のようなことを考えなければならぬ。

数値解のユークリッド・ノルムと各成分をみた場合に分けて安定性を定義し、有理型ルンゲークッタ法と線型ルンゲークッタ法の安定性を比較する。有理型ルンゲークッタ法の実用例を紹介する。

2. 有理型と線型ルンゲークッタ法の比較

解の挙動をユークリッド・ノルムと成分としてみた時の安定性の定義をそれぞれ与え、この時、演繹される性質を示す。最後に、安定性の違いを比較する。

安定性を解析するのに、テスト方程式

$$y' = Ay \quad (2.1)$$

を使い、行列 A は次のようにユニタリー行列 U と対角行列 Λ に分けられるもの考える。

$$A = \bar{U}^T \Lambda U, \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_S), \text{Re}(\lambda_i) < 0, i=1, 2, \dots, S \quad (2.2)$$

(2.1) 式を直接解析するのは面倒なので、図 1. に示したことを考える。つまり、(2.1) 式を直接ルンゲークツタ法で解いても、(2.1) 式をユニタリー変換して

$$\begin{array}{ccc} y' = A y & \xrightarrow{v = \bar{U}^T y} & v' = \Lambda v \\ y_n & & v_n \\ \downarrow \text{RK法} & & \downarrow \text{RK法} \\ y_{n+1} & \xleftarrow{y = U v} & v_{n+1} \end{array}$$

図 1. ユニタリー変換

$$v' = \Lambda v \quad (2.3)$$

(2.3) 式のように係数行列 Λ が対角行列にしてからルンゲークツタ法で解き、その解を逆ユニタリー変換して得られる値が変わらないことを考える。線型ルンゲークツタ法は明らかにユニタリー変換で不変であることが分る。ベクトル演算 (1.3) がユニタリー変換で不変であることから、有理型ルンゲークツタ法もユニタリー変換で不変であることが分る。

2.1 解のユークリッド・ノルムに関する安定性

ここでは, Hairer [2] が定義したものをそのまま使うことにする。

定義 ルンゲークツタ法による (2.3) 式の解を v_n, v_{n+1} としたとき、 S 次元複素空間

$$D = \{(z_1, z_2, \dots, z_S) \mid \|v_{n+1}\| \leq \|v_n\|\}$$

を安定領域という。ただし、 $z_i = h\lambda_i$ ($i=1, 2, \dots, S$) であり、 $\|\cdot\|$ はユークリッド・ノルムである。

定義 ルンゲークッタ公式が

- (i) $\{(z_1, z_2, \dots, z_s) \mid \operatorname{Re}(z_i) \leq 0, i=1, 2, \dots, s\} \subset D$ を満たすなら **A-安定** と呼ぶ。
- (ii) $\{(z_1, z_2, \dots, z_s) \mid |\pi - \arg(z_i)| \leq \alpha, i=1, 2, \dots, s\} \subset D$ を満たすなら **$A(\alpha)$ -安定** と呼ぶ。
- (iii) $\{(x_1, x_2, \dots, x_s) \mid x_i \leq 0, i=1, 2, \dots, s\} \subset D$ を満たすなら **A_0 -安定** と呼ぶ。
- (iv) $\{(i\omega_1, i\omega_2, \dots, i\omega_s) \mid \omega_i \text{ は実数}, i=1, 2, \dots, s\} \subset D$ を満たすなら **I-安定** と呼ぶ。

Hairer は上の定義から次の定理を導びいている。

定理 1. 有理型ルンゲークッタ法 (1.2) が

(i) A_0 -安定

(ii) $C, b_{\nu, \nu-1} \dots b_{32} b_{21} \neq 0$

を満たすならば

$$w_{ij} = 0, i+j > \nu$$

となる。

また, 有理型ルンゲークッタ 2 段公式

$$\begin{cases} y_{n+1} = y_n + h \frac{g_1 g_1}{c_1 g_1 + c_2 g_2}, & c_1 + c_2 = 1 \\ g_1 = f(y_n) \\ g_2 = f(y_n + h b_2 g_1) \end{cases} \quad (2.4)$$

について, 定理 2 を導びいている。(2.4) 式は 1 次となり,

$b_2 C_2 = -\frac{1}{2}$ ならば 2 次となる。定理 1 より, A_0 -安定となるために $w_{21} = w_{22} = 0$ とした。

定 理 2 公式 (2.4) は

- (i) $b_2 C_2 \leq -\frac{1}{2}$ ならば A_0 -安定となる。
- (ii) $b_2 C_2 \leq -\frac{1}{2 \cos \alpha (2 - \cos \alpha)}$, $\alpha \in [0, \frac{\pi}{2})$ ならば $A(\alpha)$ -安定となる。
- (iii) $b_2 C_2 \leq -\frac{1}{2}$ ならば I -安定となる。

定理 2 より, 解のユークリッド・ノルムをみた場合, 有理型ルンゲークッタ法は弱い A -安定性をもっていることが分る。

また, 線型ルンゲークッタ法についていえば, 次節 2.2 で述べるように, 解の各成分に関して安定な公式が得られるので, 明らかに, どんなノルムについても安定な公式が得られる。

2.2 解の各成分に関する安定性

ここでは, Sottas が与えた定義 [3] を一般化したものを考える。

定 義 (Sottas が与えた定義)

固有値 $\lambda_i \leq 0$ ならば, (2.3) 式の解 v_n, v_{n+1} が常に

$$\left| \frac{v_{n+1}^i}{v_n^i} \right| = \left| f_i(z_1, z_2, \dots, z_s; v_n) \right| \leq 1, \quad i=1, 2, \dots, s$$

である公式を CA_0 -安定と呼ぶ。

定義 $\operatorname{Re}(\lambda_i) < 0$ ならば, 常に, (2.3) 式の解 v_n, v_{n+1} が

$$\left| \frac{v_{n+1}^i}{v_n^i} \right| = |f_i(z_1, z_2, \dots, z_s; v_n)| \leq 1, \quad i=1, 2, \dots, s$$

である公式を **CA-安定** と呼ぶ。

定義 CA -安定で

$$\lim_{\operatorname{Re}(z_i) \rightarrow -\infty} |f_i(z_1, z_2, \dots, z_s; v_n)| = 0, \quad i=1, 2, \dots, s$$

を満たす公式を **CL-安定** と呼ぶ。

有理型ルンゲークッタ法に関しては, $Sattas[]$ により, 次の定理が導びかれている。

定理3 CA_0 -安定な有理型ルンゲークッタ法は存在しない。

線型ルンゲークッタ法に関しては, 容易に, 次の定理が導びかれる。

定理4 (2.3) 式の解の各成分の比は $z_i (= h\lambda_i)$ のみの関数となる。すなわち,

$$\frac{v_{n+1}^i}{v_n^i} = f_i(z_i), \quad i=1, 2, \dots, s$$

しかも、関数 $f_i (i=1,2,\dots,s)$ は全て同じ形をしている。すなわち、

$$f_1(z) = f_2(z) = \dots = f_s(z)。$$

定理4より、線型ルンゲークッタ法は一つの関数 $f(z)$ について A -安定であるかどうか判定できれば、微分方程式系についても成り立つことが分る。従って、単一微分方程式に関して A -安定であることを決定すれば、微分方程式系についても A -安定になることが分る。また、線型ルンゲークッタ法は、単一微分方程式について A -安定や L -安定な公式が存在することが知られている。従って、以下の定理が導びかれる。

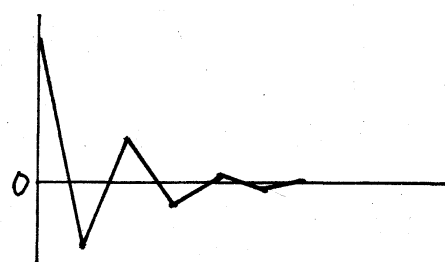
定理5 CA -安定や CL -安定な線型ルンゲークッタ法が存在する。

2.3 解の挙動の比較

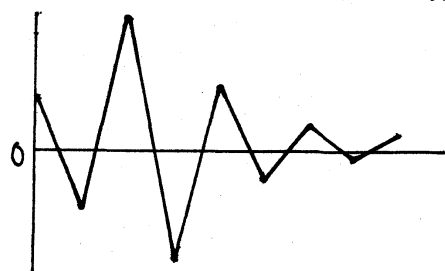
2.1節と2.2節から、有理型ルンゲークッタ法は解のユークリッド・ノルムに関しては安定な公式が得られるのに対し、線型ルンゲークッタ法は解のどんなノルムに関しても、各成分に関しても安定な公式が得られる。この節では、安定性を解のノルムで決めた場合と各成分で決めた場合の解の挙動の違い

いを比較する。

安定性をノルムで定義したのと、成分で定義した結果に大きな違いがでてくるのは、初期値の大きさが非常に異なるときである。安定性を成分で定義できれば、初期値の大きさに関係なく、図2(1)のように、解は単調に収束する。安定性をノルムで定義した場合は、初期値の大きさが同じ位なら、解は単調に収束してくれる。しかし、初期値の大きさが非常に異なると、初期値が大きい解はノルムといっしよに単調に収束するが、初期値が非常に小さい解は少し大きくなってもノルムに影響を与えないので図2(2)のような収束の仕方もある。



(1) 解が単調に収束する場合



(2) 解が単調でなく収束する場合

図2. 解の収束

例

$$y' = \begin{pmatrix} -1 & 0 \\ 0 & -4 \end{pmatrix} y, \quad y_0 = \begin{pmatrix} 1 \\ 0.1 \end{pmatrix}$$

を A(2.97)-安定な有理型ルンゲークッタ2段1次公式

$$\begin{cases} y_{n+1} = y_n + h \frac{g_1 g_2}{2g_1 - g_2} \\ g_1 = f(y_n) \\ g_2 = f(y_n + h g_1) \end{cases} \quad (2.5)$$

ステップ幅 $h=1$ で解いた結果を図3に示す。図3より、 y_1 とユーッド・ノルムは単調に収束していることが分る。 y_2 は $x=1$ のところで一旦増加してから収束していることが分る。

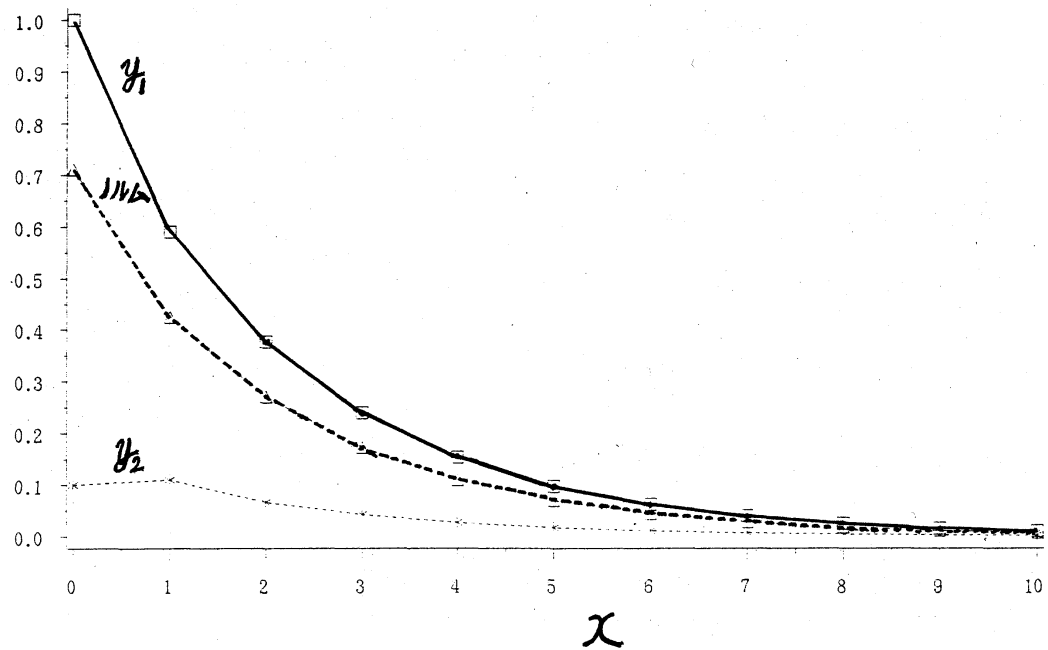


図3(1) 例題の数値解

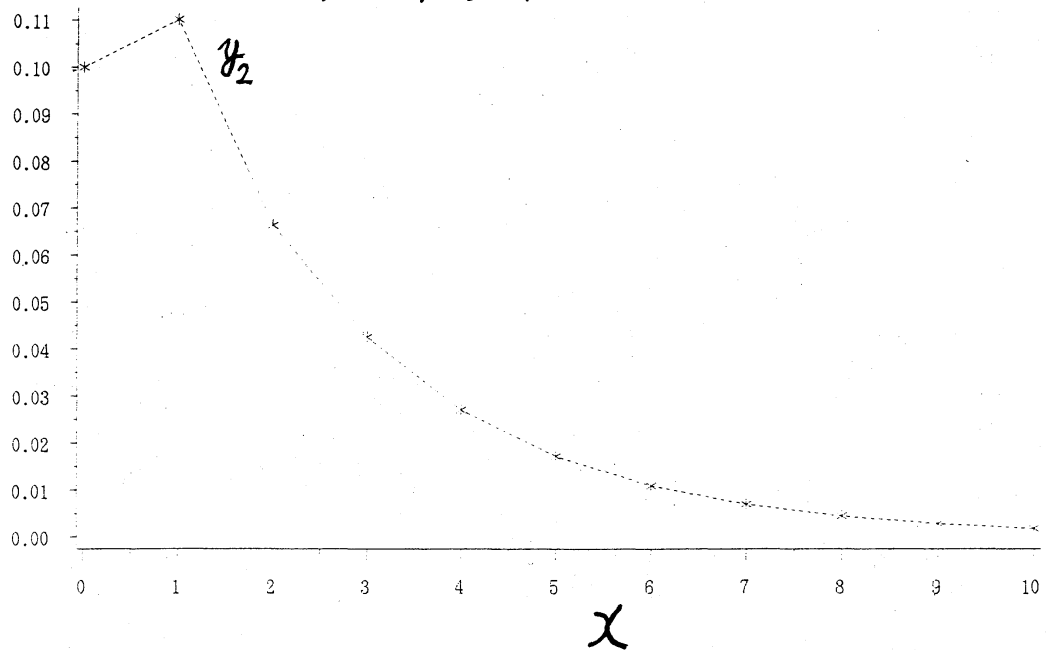


図3(2) 例題の y_2 の拡大図

3. 有理型ルンゲークッタ法の実用例

放物線型偏微分方程式の数値解法の一つとして、方程式の一部を差分近似して常微分方程式に直してから解く方法がある。こうすると、硬い微分方程式系となることが知られている。線型陰的ルンゲークッタ法は安定性がよいが、計算の途中に連立方程式を解くなど複雑な過程が必要である。有理型ルンゲークッタ法は安定性がよいし、陰的な方法ほど計算の途中に複雑な過程がない。有理型ルンゲークッタ法は手軽に使えるので、放物線型偏微分方程式の数値解法によく使われている。有理型ルンゲークッタ法の拡散移流方程式と *implicit-explicit time integration* への応用を紹介する。

例1. 線形拡散移流方程式

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - 25 \frac{\partial u}{\partial x}, & 0 \leq x \leq 1, t \geq 0 \\ u(0, t) = 1, \frac{\partial u}{\partial x}(1, t) = 0, & t > 0 \\ u(x, 0) = 0, & 0 < x < 1 \end{cases} \quad (3.1)$$

の数値解法を考える。ただし、境界条件を $\frac{\partial u}{\partial x}(1, t) = 0, t > 0$ としたように、外部と絶縁されていて熱の出入りはないものとする。(3.1) 式を中間差分近似

$$\begin{cases} x_i = ih, h = \frac{1}{n} \quad (i=1, 2, \dots, n) \\ u_i(t) = u(x_i, t) \\ \frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \quad \frac{\partial u}{\partial x} \approx \frac{u_{i+1} - u_{i-1}}{2h} \end{cases}$$

で常微分方程式系

$$\frac{d}{dt} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} = \begin{pmatrix} -2 & 1-12.5h & & & 0 \\ 1+12.5h & -2 & 1-12.5h & & \\ & \ddots & \ddots & \ddots & \\ 0 & & 1+12.5h & -2 & 1-12.5h \\ & & & 1 & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} + \begin{pmatrix} 1/h^2 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} u_1(0) \\ u_2(0) \\ \vdots \\ u_{n-1}(0) \\ u_n(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (3.2)$$

に直す。(3.2)式を次の2つの方法RAT, LSTIFF2で解く。ただし, $N=20$, 初期ステップ幅 $h_0=0.00022$, 積分区間 $[0, 100]$ とする。
[RAT]有理型ルンゲ-クッタ法2(3)次埋め込み公式[2]

A₀-安定な2段2次公式

$$\begin{cases} y_{n+1} = y_n + h \frac{g_1 g_1}{2g_1 - g_2} \\ g_1 = f(y_n) \\ g_2 = f(y_n + \frac{1}{2}h g_1) \end{cases} \quad (3.3)$$

(3.3)式に埋め込む3次公式

$$\begin{cases} \tilde{y}_{n+1} = y_n + h \frac{\frac{1}{4}g_1 g_1 + \frac{1}{2}g_1 g_3 + \frac{1}{4}g_3 g_3}{\frac{2}{3}g_1 - \frac{1}{6}g_2 + \frac{1}{3}g_3 + \frac{1}{6}g_4} \\ g_3 = f(y_{n+1}) \\ g_4 = f(y_{n+1} + \frac{1}{2}h g_3) \end{cases} \quad (3.4)$$

(3.3) 式の局所離散化誤差 $TE = \tilde{y}_{n+1} - y_{n+1}$ とおくと,

$$TE = \frac{3}{2} \frac{(y_1 - y_3)(y_1 - y_3)}{(4y_1 - y_2 + 2y_3 + y_4)} - \frac{y_1 y_1}{2y_1 - y_2} \quad (3.5)$$

と書ける。これを使って、ステップ幅を調節をする。(3.5)式から誤差のノルム ERR を求める。

$$ERR = \left[\sum_{i=1}^5 \{ |TE_i| / (1 + |y_{n+1}^i|)^2 \} \right]^{1/2}$$

与えられた許容誤差を TOL としたとき、新しいステップ幅は

$$h_{new} = h_{old} \cdot \max \{ 0.5, \min(1.5, 0.9 \sqrt[3]{\frac{TOL}{ERR}}) \} \quad (3.6)$$

とする。このとき、次の規則に従って積分する。

(i) $ERR > TOL$ のとき、ステップ幅 h_{new} でもう一度そのステップを計算する。

(ii) $ERR \leq TOL$ のとき、そのステップでの計算値を受け取り、新しいステップ幅で次のステップを計算する。

[LSTIFF2] L-安定な2段2次Rosenbrock法[1]

$$\begin{cases} y_{n+1} = y_n + h k_2 \\ k_1 = A(y_n)^{-1} f(y_n) \\ k_2 = A(y_n)^{-1} f(y_n + h(\sqrt{2}-1)/2 k_1) \\ A(y_n) = I - (1-\sqrt{2}/2) h \frac{\partial f}{\partial y}(y_n), \quad I \text{ は単位行列} \end{cases} \quad (3.7)$$

y_{n+1} を y_n からステップ幅 h で計算した値とし、 y_{n+1}^* を y_n からステップ幅 $\frac{h}{2}$ で2度計算した値とする。局所離散化誤差 E_{n+1}

$$E_{n+1} = \|y_{n+1}^* - y_{n+1}\| / 3$$

で与えられる。このとき、次の規則に従って積分する。

- (i) $E_{n+1} > TOL$ のとき, 期待誤差 $\approx \frac{TOL}{5}$ となるように新しいステップ幅を決め, もう一度そのステップを計算する。
- (ii) $\frac{3}{4}TOL < E_{n+1} \leq TOL$ のとき, 期待誤差 $\approx \frac{TOL}{5}$ となるように新しいステップ幅を決め, 次のステップに移る。
- (iii) $\frac{1}{10}TOL < E_{n+1} \leq \frac{3}{4}TOL$ のとき, ステップ幅をそのままにして, 次のステップに移る。
- (iv) $E_{n+1} \leq \frac{1}{10}TOL$ のとき, 期待誤差 $\approx \frac{TOL}{2}$ となるように新しいステップ幅を決め, 次のステップに移る。

この結果は表1, 表2, 図4に示す。表1は2つの方法で $t=100$ まで積分したときの計算回数の比較を示す。ただし,

FUN: 関数の計算回数, JAC: ヤコビアン の計算回数

LUD: LU分解 の回数, SOL: 連立1次方程式を解いた回数

STE: ステップ数, REJ: 捨てたステップ数

とする。RATの方が計算回数が少ないことが分る。表2は, $t=100$ での数値解の比較を示した。RATは許容誤差と同程度の精度であるのに対し, LSTIFF2は許容誤差に無関係に高精度になっていることが分る。図3は許容誤差と計算時間の関係を示した。表1, 表2の結果からも明らかなように, RATの方が効率的に解けることが分る。

表1. 計算回数

TOL = 10^{-3} の場合

	FUN	JAC	LUD	SOL	STE	REJ
RAT	376	—	—	—	92	2
LSTIFF2	2525	1010	1515	3030	503	2

TOL = 10^{-5} の場合

	FUN	JAC	LUD	SOL	STE	REJ
RAT	976	—	—	—	241	3
LSTIFF2	24765	9906	14859	29718	4952	1

表2. $t=100$ での u_{20} の数値解

TOL	RAT	LSTIFF2
10^{-1}	1.000012228653061	0.999999999999997132
10^{-2}	0.9999998992547525	0.99999999999999542
10^{-3}	0.9999971400662994	0.99999999999999948
10^{-4}	1.000000060455869	0.99999999999999995
10^{-5}	0.9999999981257375	0.99999999999999962

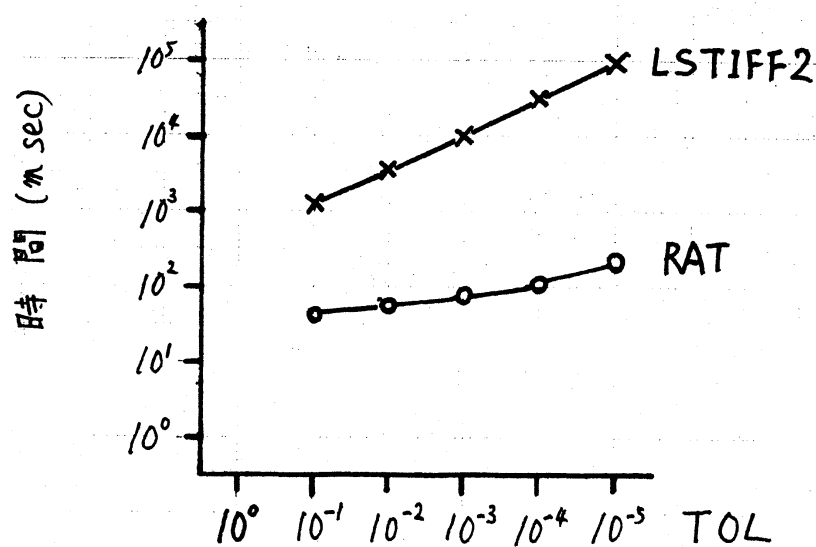


図4. 許容誤差に対する計算時間

例2. 熱伝導率が異なる材料をつないだときの熱伝導

熱伝導率が異なる材料をつないだとき、熱伝導方程式の数値解法について考える。それぞれの材料のところで、要求される公式の安定性が異なる。同じステップ幅で解こうとすると、最も安定性が強い公式を用いなければならない。しかし、安定性をそれほど必要としない所を強い安定性をもつ公式で解くのも不経済である。そこで提案されたのが *implicit-explicit time integration* という方法である。この方法は、安定性を必要としない所では陽的な方法を使い、安定性を必要とする所では陰的な方法を使うというものである。ここでは、上記の方法の陰的な方法を使う部分を有理型ルンゲークッタ法と置きかえたものと比較する。

次の熱伝導方程式を考える。

$$\begin{cases} \frac{\partial u}{\partial t} = \rho(x) \frac{\partial^2 u}{\partial x^2}, & \rho(x) = \begin{cases} 0.1 & (0 \leq x \leq 1) \\ 0.4 & (1 < x \leq 2) \end{cases} \\ u_0 = u(0, t) = \begin{cases} 0.4t & (0 \leq t \leq 25) \\ -0.4t + 20 & (25 \leq t \leq 50) \\ 0 & (t > 50) \end{cases}, & \frac{\partial u}{\partial x}(2, t) = 0 \\ u(x, 0) = 0, & x > 0 \end{cases} \quad (3.8)$$

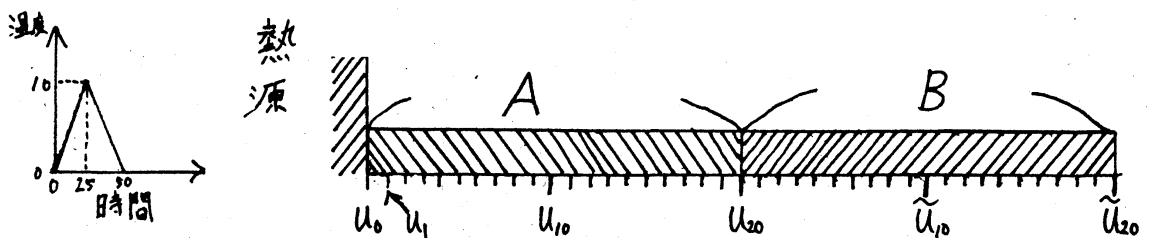


図5. 熱伝導率の異なるものの熱伝導

ただし, 外部とは絶縁されているものとする。図5のA, Bの部分をそれぞれ中間差分近似

$$A: \begin{cases} x_i = i\Delta x, \quad \Delta x = \frac{1}{n} \quad (i=1, 2, \dots, n) \\ u_i(t) = u(x_i, t) \\ \frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} \end{cases} \quad B: \begin{cases} x_i = i\Delta x + 1, \quad \Delta x = \frac{1}{n} \quad (i=1, 2, \dots, n) \\ \tilde{u}_i = u(x_i, t) \\ \frac{\partial^2 u}{\partial x^2} \approx \frac{\tilde{u}_{i-1} - 2\tilde{u}_i + \tilde{u}_{i+1}}{\Delta x^2} \end{cases}$$

して, (3.8)式を常微分方程式系

A:

$$\frac{d}{dt} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} = \frac{\rho(x)}{\Delta x^2} \begin{pmatrix} -2 & 1 & & 0 \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ 0 & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{pmatrix} + \frac{\rho(x)}{\Delta x^2} \begin{pmatrix} u_0 \\ 0 \\ \vdots \\ 0 \\ \tilde{u}_1 \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \\ \vdots \\ u_{n-1}(0) \\ u_n(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (3.9)$$

B:

$$\frac{d}{dt} \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_{n-1} \\ \tilde{u}_n \end{pmatrix} = \frac{\rho(x)}{\Delta x^2} \begin{pmatrix} -2 & 1 & & 0 \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ 0 & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix} \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_{n-1} \\ \tilde{u}_n \end{pmatrix} + \frac{\rho(x)}{\Delta x^2} \begin{pmatrix} u_n \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} \tilde{u}_1(0) \\ \tilde{u}_2(0) \\ \vdots \\ \tilde{u}_{n-1}(0) \\ \tilde{u}_n(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (3.10)$$

に直す。(3.9), (3.10)式を一般化したものを

$$\begin{cases} \dot{v} = K\theta + f & \frac{d\theta}{dt} = v \\ \theta(0) = \theta_0 \end{cases} \quad (3.11)$$

と書くことにする。次の2つの方法を比較する。

[implicit-explicit time integration]

Δt を時間方向のステップ幅とし, α を公式の安定性と精度

を決めるパラメータとする。

• explicit

$$\begin{cases} v_{n+1} = K \tilde{\theta}_{n+1} + f_{n+1} \\ \tilde{\theta}_{n+1} = \theta_n + (1-\alpha) \Delta t v_n \\ \theta_{n+1} = \tilde{\theta}_{n+1} + \alpha \cdot \Delta t \cdot v_n \end{cases} \quad (3.12)$$

• implicit

$$\begin{cases} v_{n+1} = K \theta_{n+1} + f_{n+1} \\ \tilde{\theta}_{n+1} = \theta_n + (1-\alpha) \Delta t v_n \\ \theta_{n+1} = \tilde{\theta}_{n+1} + \alpha \cdot \Delta t v_{n+1} \end{cases} \quad (3.13)$$

[RRK-explicit time integration]

上の公式で(3.12)式は同じ, (3.13)式を次の公式とかわえたものを考える。

• modified RRK (有理型ルンゲークッタ法)

$$\begin{cases} v_1 = K \theta_n + f_{n+1}, & v_2 = K(\theta_n + \frac{1}{2} \Delta t v_1) + f_{n+1} \\ b = 2v_1 - v_2, & e = 2(v_1^T b)v_1 - (v_1^T v_1)b \\ \theta_{n+1} = \theta_n + \Delta t e / (b^T b) \end{cases} \quad (3.14)$$

ここでは, $\alpha=1$, $n=20$ を考える。 $\alpha=1$ とすると, explicitは Euler methodとなり, implicitは backward Euler Method となる。 Euler methodの解が安定となるためには

$$\Delta t < \frac{2}{|\lambda_{\max}|}, \quad \lambda_{\max} \text{は } A \text{で最大の固有値}$$

なので, $\Delta t = 0.0125$ で積分する。

結果としては, 2つの方法とも数値解は発散することなく安定している。しかし, 有理型ルンゲクッタ法を使った方法は山のピークが少し遅れてでており, 精度の面では *implicit* に比べて劣ることが分る。

4. おわりに

有理型ルンゲクッタ法は, 積分の始めの部分で誤差が大きくなることを除けば, 硬い微分方程式系を解くのに手軽に使える方法である。

参考文献

- [1] Bui, T.D. and Bui, T.R.: Numerical Methods for Extremely Stiff Systems of Ordinary Differential Equations, Appl. Math. Modelling, Vol.3, pp.355~358 (1979).
- [2] Hairer, E.: Unconditional Stable Explicit Methods for Parabolic Equations, Numerische Mathematik, Vol.35, pp.57~68 (1980).
- [3] Sattas, G.: Rational Runge-Kutta Methods Are Not Suitable for Stiff Systems of ODEs, Journal of Computational and Applied Mathematics, Vol.10, pp.169~174 (1984)
- [4] Wombecz, A.: Rational Runge-Kutta Methods for Solving Systems of Ordinary Differential Equations, Computing, Vol.20, pp.333~342 (1978)